IMPLEMENTATION OF SCHEDULING TECHNIQUES FOR FOG COMPUTING (USING JAVA)

Project report submitted in partial fulfillment of the requirementfor the degree of Bachelor of Technology

in

Computer Science and Engineering/Information Technology

By

Arj Srivastava (191390)

Under the supervision of

Dr. Diksha Hooda

to

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology Waknaghat, Solan-173234, Himachal Pradesh

CANDIDATE’S DECLARATION

I hereby declare that the work presented in this report entitled “**Implementation of Scheduling Algorithm for Fog Computing** ” in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering/Information Technology** submitted in the department of Computer Science/Engineering and Information Technology, Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from August 2022 to May 2023 under the supervision of **Dr. Diksha Hooda (Assistant Professor (SG), Dept. CSE & IT)**. I also authenticate that I have carried out the above-mentioned project work under the proficiency stream of **Cloud Computing**. The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Arj Srivastava, 191390

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Dr. Diksha Hooda

Assistant Professor (SG)

Department CSE & IT

Dated:

ACKNOWLEDGEMENT

Firstly, I express my heartiest thanks and gratefulness to Almighty God for his divine blessing that makes it possible to complete the project work successfully.

I am really grateful and wish my profound indebtedness to Supervisor **Dr. Diksha Hooda, Assistant Professor (SG),** Department of CSE Jaypee University of Information Technology, Wakhnaghat. Deep Knowledge & keen interest of my supervisor in the field of "**Cloud Computing and Fog Computing**" to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting them at all stages have made it possible to complete this project.

I would like to express my heartiest gratitude to **Dr. Diksha Hooda**, Department of CSE, for his kind help to finish my project.

I would also generously welcome each one of those individuals who have helped me straightforwardly or in a roundabout way in making this project a win. In this unique situation, I might want to thank the various staff individuals, both educating and non-instructing, which have developed their convenient help and facilitated my undertaking.

Finally, I must acknowledge with due respect the constant support and patience of my parents.

Project Group No.: 89
Student Name: Arj Srivastava

Rollno.: 191390

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

1. WSN: Wireless Sensor Networks.
2. IoT: Internet of Things.
3. LED: Light-Emitting Diode.
4. LTE: Long-Term Evolution.
5. RFID: Radio-Frequency Identification.
6. CCTV: Closed-Circuit Television.
7. WiFi: Wireless Fidelity.
8. API: Application Programming Interface.
9. DCNN: Deep Convolutional Neural Network.
10. FCFS: First Come-First Served.
11. CPU: Central Processing Unit.
12. RGB: Red Blue Green.

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

In this project, we employed the notion of a smart vehicle parking system to create the fog - based scheduling method. Personal vehicles are becoming a major mode of mobility, and the automotive sector is expanding daily. Finding a free parking space for one's automobile has become more challenging as rural road transit has increased, but has also resulted in a variety of issues, including: Congestion, wastage of time, eutrophication, and, most critically, excessive energy utilization.

On the other hand, the number of car outlets in regions is not growing at same rate as number of automobiles on the roads. Intelligent parking solutions have become essential as a consequence in order to overcome the primary obstacles. Several researchers have sought to use slashing technology to streamline parking facilities distribution.

Few researchers have created intelligent parking technologies that employ cloud computing technologies. On the contrary side, latency is a serious problem for cloud-based programs, particularly autonomous parking solutions and linked automobiles. Fog computing, that exposes cloud - based applications near to the ip layer, overcomes the delay issue while also providing important benefits such as rapid flexibility, energy portability, and safety.

Fog computing is primarily used in the suggested technique to reduce latency and resource use among the autonomous parking management systems. Experiments in iFogSim were done to illustrate the success of the proposed technique for cutting latency and network consumption, and the outcomes were juxtaposed to those of the advanced ai parking service's virtualized deployment.

The simulation results demonstrate that proposed fog-based operation of optimal traffic management system decreases latency considerably. Furthermore, as compared to digital deployment of automatic parking zones, the proposed fog-based approach reduces total transmission consumption.

CHAPTER 1: INTRODUCTION

1.1 Introduction

The rapid urbanization of the world has resulted in a surge in the number of people living in cities, leading to overcrowding, and an increase in the utilization of automobiles for commuting to work. Consequently, parking spots in urban areas have become scarce, and finding a parking spot during peak hours has become a challenging task. This has led to traffic bottlenecks, time loss, fuel usage, and environmental degradation.

According to studies, locating a parking space in America alone costs around 740 tons of Carbon dioxide, 94,000 hours, and 48,000 litres of petrol. As the urbanization trend continues, congestion is likely to worsen, and the number of automobiles is expected to increase, making the problem even more severe.

To address this issue, the concept of a smart city has emerged, which integrates information and communication technologies (ICT) to facilitate sustainable development, economic growth, and quality-of-life improvement. One of the key areas leading towards smart cities is the development of intelligent transportation and efficient traffic management systems to optimize drivers' travel planning and alleviate traffic congestion.

Currently, drivers search for parking spots by themselves, wasting valuable time that could hinder economic growth. Moreover, during congestion, vehicles burn fuel unnecessarily, leading to high greenhouse gas emissions. These issues can be mitigated by implementing a smart parking system that assists drivers in determining and reserving parking spots in advance before reaching their destination.

Such systems would become highly significant in almost every major city in the world, especially during large events like conferences, religious events, sporting events, festivals, and concerts. Smart parking systems form part of traffic management strategies and manage parking processes by helping vehicles park efficiently in a manner that reduces parking arrival time and traffic congestion.

The development of smart parking systems has attracted the attention of researchers in academia and industry due to their economic, environmental, and aesthetic impacts. The Internet of Things (IoT) has emerged as a leading technology for smart object communication, and it is the key enabler technology for smart parking systems that employ different sensing technologies (e.g., ultrasonic, magnetometers, and visual sensors) and networking technologies (e.g., ZigBee, Wi-Fi, and cellular).

It facilitates real-time monitoring and control by collecting real-time sensor data and integrating with other technologies such as machine learning, sensor fusion, and computing. Several commercial parking systems have been developed, such as SmartParking, PlacePod, and Sitraffic Scala, which offer various features like remote parking-spot booking, fee payment, interactive parking maps, and more.

However, the main drawback of current systems is their high cost and limited development capability for public developers since they are not open source. In addition, the current systems do not meet today’s parking needs and the expectations of smart city evolution.

Therefore, it is necessary to develop open-source smart parking systems that meet current parking needs and are adaptable to future developments. The development of such systems will help reduce traffic congestion, pollution, and negative impacts on the city landscape and environment, promoting sustainable development and improving the quality of life for city dwellers.

1.2 Problem Statement

In this project, we have implemented a smart vehicle parking system using the fog computing methodology. The issue of parking difficulties has gained significant attention in recent times due to the increasing number of cars on the road. As a result, numerous researchers have developed automated car parking solutions that rely on cloud-based services. While these smart vehicle parking systems have introduced cutting-edge technology to this field, the widespread real-time implementation of these solutions still remains a challenge.

One of the possible impediments to the widespread adoption of cloud-based virtualized solutions for smart auto parking is the high latency rate and network usage. For instance, in a virtualized solution proposed in [11], the parking places' status is updated in the cloud after employing a technique to quantify parking spaces. However, a significant problem was identified when connecting the sensors to the cloud, which resulted in a firm latency that is unsuitable for applications that require fast computational time. The inability of the technique to update the information in the cloud whenever a car enters or exits a specific parking space is a major issue.

Fog computing has emerged as a potential solution to address the latency and network usage issues associated with cloud-based virtualized solutions. Fog computing involves decentralizing the computing resources and distributing them closer to the network's edge, which can significantly reduce latency and improve the response time for real-time applications such as smart vehicle parking systems.

By using fog computing, we have been able to overcome the latency and network usage issues faced by cloud-based virtualized solutions. In our smart vehicle parking system, the sensors are connected to the fog nodes, which are located at the network's edge, rather than the cloud. This allows for real-time data processing and quick response times, ensuring that the parking spaces' status is updated immediately whenever a car enters or exits a specific parking space.

Overall, the implementation of fog computing in our smart vehicle parking system has enabled us to achieve real-time data processing and overcome the latency and network usage issues faced by cloud-based virtualized solutions. As a result, we can offer a reliable and efficient solution to the parking difficulties faced by drivers in big cities..

1.3 Objective

The project aims to explore the concepts of fog and cloud computing and their applications in the field of process scheduling. Specifically, it aims to develop a scheduling scheme using Java that can predict working data buildup on both fog nodes and the cloud. The goal is to create an efficient backend mechanism for an intelligent parking system for cars.

Fog computing and cloud computing are both models for delivering computing resources and services over a network. Cloud computing involves delivering computing resources, such as servers, storage, and applications, over the internet. In contrast, fog computing involves distributing computing resources and services closer to the network's edge, such as on local servers or routers. This allows for faster data processing and response times, making it ideal for real-time applications.

To optimize the scheduling scheme, the project will leverage both cloud and fog computing resources. The scheduling scheme will predict the buildup of working data on both fog nodes and the cloud and allocate resources accordingly to minimize delays and optimize performance. By analyzing the delay in transmitting and analyzing data, the project will determine which approach, fog or cloud computing, is superior for the intelligent vehicle parking system.

To achieve these goals, the project will use an intelligent vehicle parking simulation environment. This environment will simulate the conditions of a real-world parking system and provide data on the performance of the scheduling scheme. By analyzing the data, the project will identify areas for improvement and refine the scheduling scheme to achieve optimal performance.

Overall, the project aims to explore the use of fog and cloud computing in process scheduling and develop an optimized scheduling scheme for an intelligent vehicle parking system. By leveraging both cloud and fog computing resources, the project aims to create a fast-working backend mechanism that can efficiently manage the parking system. Through analysis and refinement, the project will determine which approach, fog or cloud computing, is superior for the system, providing valuable insights into the optimal use of these technologies.

1.4 Methodology

The current study proposes a fog-based intelligent parking facilities design to address the issues associated with cloud-based approaches for parking management systems. This proposed design is unique in that it employs fog computing as an interim phase in a tri fog-based proposal for an ideal car parking control system.

The fog server is responsible for assessing the parking photos that are accessible, and the condition of the parking details is displayed on the LED. If a parking place doesn't have any available parking spots, the parking area's LED shows details about neighboring parking spaces. This feature helps drivers to identify available parking spots quickly, even when the nearest parking space is occupied.

Furthermore, the proposed fog-based framework has been developed with the objective of minimizing latency and bandwidth utilization. By minimizing latency and resource utilization, the driver can select a parking spot and leave the car there without wasting too much time, petrol, or Carbon dioxide. Additionally, the involvement of fog nodes enhances the efficiency of discovering unoccupied parking bays. If a parking spot is not available at a particular parking location, the system can provide information about nearby open parking garages, thereby reducing turnaround time for parking.

The effectiveness and productivity of the suggested fog-based vehicle parking idea have been evaluated through comprehensive simulations. The experiment findings reveal a considerable reduction in latency and network use when compared to the cloud-based performance of the intelligent parking platform. This finding indicates that the proposed fog-based system is more efficient and effective in managing parking spaces and ensuring a hassle-free parking experience for drivers.

Overall, the proposed fog-based intelligent parking facilities design offers a unique solution to the challenges associated with cloud-based parking management systems. By employing fog computing as an interim phase in the tri fog-based proposal, the system can deliver real-time parking information, minimize latency and bandwidth utilization, and provide drivers with a more efficient and effective parking experience. The comprehensive simulations indicate that the proposed system is an effective alternative to traditional cloud-based systems and is a valuable contribution to the field of intelligent parking management.

CHAPTER 2: LITERATURE SURVEY

2.1 Background

Although cloud computing is widely used, it has some drawbacks, such as higher response times and increased access fees. Researchers have therefore turned to fog computing, which has shown promise in reducing processing delays and network utilization issues. Aazam et al. [1] compared cloud-based and fog-based systems and found that fog significantly reduced processing delay. By moving some modeling and analysis duties from the cloud platform to the fog layer, fog computing is seen as an excellent way to meet the needs of many connections and diverse network capabilities. Fog computing has shown great potential for decreasing cloud computing delay and network utilization issues, and numerous research studies have looked into fog-based solutions in a variety of industries to increase control performance, protection, and economic viability.

Vilalta et al. [2] proposed a fog-based infrastructure that provides network topologies such as hybridization, portable cloud technologies, and virtualization to the channel's outer limit. This enhances the flexibility of the system. Bi et al. [3] proposed a fog computing infrastructure based on applications and services connectivity to provide portable consumers with clear and perfect transportation assistance. Additionally, the researchers developed an effective signaling action to optimize the pathway for the transportation system.

Fog computing has the substantial benefit of lowering the rate of information transfer to the cloud, reducing network use by apps communicating with the server, and improving the success rate of the system. Fog computing has been shown to have better connectivity than cloud computing in some instances. Furthermore, fog computing reduces network traffic while promoting mobility, making it an ideal solution for IoT platforms.

Network utilization is often a key component of practical systems, and it can be effectively decreased using fog computing. Passas et al. [4] introduced a multi-heterogeneous wireless technology system that offers a remedy for the Paris subway rate plan supply experience to the users who do not want to be disorganized by preserving the same wagons for different fares. Miliotis et al. [5] presented a balanced proportionately fair spectrum access mechanism and a constant rate delivery system for the balance storage capacity supplied through LTE uplink to decrease capacity problems. Fog nodes have a high throughput and are widely dispersed.

In the proposed intelligent vehicle parking system, image recognition and fog computing are used to identify the status of the parking spaces when a new car arrives at the parking lot. An image of the car is taken and examined for a variety of potential vehicle locations, including whether the parking space is going to be occupied, occupied, or leaving. A LED monitor is placed at the entrance gate to alert drivers to the availability of open parking spaces. Once a person parks their car in a slot, the status of the slot changes.

Private cloud computing, which emphasizes completing calculations, can be used to solve the aforementioned problems efficiently near system connections, avoiding periodic information exchange to the cloud. The proposed fog-based system considers latency and bandwidth use and improves efficiency in discovering unoccupied parking bays. The fog-based infrastructure is based on applications and services connectivity, and the researchers have proposed a pathway optimizer and an effective signaling action to provide portable consumers with clear and perfect transportation assistance.

Comprehensive simulations were used to evaluate the effectiveness and productivity of the proposed fog-based vehicle parking system. The experiment findings revealed a considerable reduction in latency and network use when compared to the intelligent parking platform's cloud-based performance. This study contributed three things: the use of fog computing as an interim phase in a tri fog-based proposal for an ideal car parking control system, the consideration of latency and resource utilization to minimize time and cost, and the assessment of the system's effectiveness and productivity using comprehensive simulations. Overall, fog computing has tremendous potential for solving many of the challenges associated with cloud computing and improving the efficiency of a wide range of applications, including intelligent vehicle parking systems.

2.2 Related Work

In recent years, there have been several studies exploring the use of technology and digital assistants to aid in parking automobiles. One such study by Ji et al. [6] proposed a cloud-based intelligent parking solution for smart cities. The methodology involved three layers: application layer, communication layer, and sensor layer.

The application layer connected many automotive parking activities to the resource center via cloud computing services, such as the parking tracking system, the parking tracking service, and the information system. The intelligence center provided access to records, and an IoT comprehensive service site was connected to the IoT control center. A comprehensive platform managed the metropolitan area, and the connection layer allowed connectivity between the detector and application levels.

The detector layer, the top layer, was composed of CCTV cameras, and the biosensor layer was coupled to the network layers. However, due to its reliance on sensors, this method was only applicable in smart cities.

On the other hand, another proposed approach involved using high-definition cameras to capture photographs of parking spaces, which were then transferred to the fog by Wi-Fi via a microprocessor. This technique was presented in a paper by an unknown author or group, which was not specified in the given text.

Researchers in [11] developed a virtualized framework for intelligent parking. They used an ultrasonic sensor, a Microcontroller, a virtual machine, and a smartphone app to create the proposed architecture. The system was configured such that twelve detectors were attached to the Microcontroller Board. The gadget was recognized via the Web Application, and the slot's value was established by the detector code. Information was then examined and applied to the parking system. An administrator interface was created to retrieve intelligent parking information and estimate available parking spaces. The site could also detect malfunctioning equipment. The data was sent to the smartphone app, where the user could examine and validate parking spot information.

However, this architecture's capacitive scanner's range was only 0.02m to 0.4 m, and a person required a mobile phone to view the information. The method was useless if the person did not possess a smartphone. Additionally, due to the internet, other issues such as delay could develop.

In [12], the literature described a computational imagination technique that used DCNN to locate unclaimed parking bays in any parking site. Although the strategy was useful for discovering parking spaces, it lacked any associated architecture that could help vehicles locate parking spaces at adequate parking entrance points.

The inventors of the proposed approach used cloud services for smart auto parking. The driving component used the intelligent system to locate the vehicle. A smartphone application was created for the individual to use this technology. The detector evaluated the condition of parking areas, and an engine located the closest parking place depending on the customer's present location. When a user logged in, the system recognized the user's current location and made a server request for information. If the system detected an empty parking space, it told the user. The parking space was reserved by the user, and the appointment was saved in the system.

However, as stated in [12], the detector would only recognize the presence of the car if it was correctly parked. Instead, the detector would not recognize the vehicle, and the parking garage would be displayed as "Free." The sensor network detected obstacles and operated at 0.5 volts. The result was transmitted to the fog node when an obstruction was identified. The disadvantage of this technique was that the driver must have an intelligent Android smartphone to obtain data about free parking places. The system used a sensor network, and productivity difficulties arose as a consequence of sensor network limitations.

However, the system had a problem, except that the detector could only gather details concerning the blockage if the automobile was properly parked. Additionally, the detectors' effectiveness was impacted by the vehicle's location and orientation

Table 1: List of Literature Review

Author(s)	**Published By**	**Methodology**	**Disadvantage**
Kamran Sattar Awaisi, et al	IEEE	Moving Towards an Efficient Car Parking Architecture with Fog	The suggested research's use of cameras to detect parking spaces has certain limitations.
Chaogang Tang, Xianglin Wei, Chunsheng Zhu et al	IEEE	Towards Smart Parking Based on Fog Computing	Fog nodes lack efficient mechanism to cope with case that response latency is very long due to the large number of parking requests with same preferences.
Mohammed Balfaqih, Waheb Jabbar et al	MDPI	Design and Development of Smart Parking System Based on Fog Computing	FCFS algorithm was used as primary method to sort the problem
Mohammed Aazam et al.	IEEE	Fog Computing Architecture, Evaluation, and Future Research Directions	Fog is quite useful in geographically dispersed areas where connectivity can be irregular.
Yi-Chieh Peter Chang et al	IEEE	Fog Computing Node System Software Architecture and Potential Applications for NB-IoT Industry	A Fog node can be viewed as a reduced capability of a cloud server and/or integrated with a small cell for handling the telecommunication services

R. Vilalta et al	IEEE	TelcoFog: A Unified Flexible Fog and Cloud Computing Architecture for 5G Networks	TelcoFog's key benefits are the dynamic deployment of new distributed low-latency services.
Y. Bi, G. Han, et al	IEEE	Mobility Support for Fog Computing: An SDN Approach	In this article, we propose a novel software-defined-networking-based fog computing architecture by decoupling mobility control and data forwarding
R. Mahmud et al	Spirnger Link	Fog Computing: A Taxonomy, Survey and Future Directions	Not proposed based on the observations, we propose future directions for research
Harshit Gupta et al	Research Gate	iFogSim: A Toolkit for Modeling and Simulation of Resource Management Techniques in Internet of Things, Edge and Fog Computing Environments	Scalability of the simulation toolkit in terms of RAM consumption and execution time is verified under different circumstances.
V. Miliotis et al	IEEE	Mobility Support for Fog Computing: An SDN Approach	Numerical results from extensive simulations have demonstrated that the proposed scheme can not only guarantee service continuity

iFogSim is a widely used simulation tool for modeling and evaluating fog computing systems. Fog computing is a distributed computing paradigm that extends cloud computing to the edge of the network, where data is generated and consumed, to provide low-latency, location-aware, and context-aware services. The iFogSim toolset allows researchers and practitioners to simulate fog computing scenarios and evaluate the performance of fog-based technologies in terms of network utilization, latency, energy consumption, and other metrics.

Several journal articles have utilized iFogSim to assess the effectiveness of fog-based technologies. For instance, Shurman and Aljarah [7] compared the cooperative approach with the benchmark methodology in iFogSim and evaluated network utilization and total latencies. Zohora et al. [8] developed Fog Systems, Sensor, Physiological Topology, and Activator classes using iFogSim. Mahmud et al. [10] reported that iFogSim is one of the most useful simulation resources available for simulating fog conditions. They utilized iFogSim to evaluate the performance of a multi-layer fog architecture in terms of response time, throughput, and energy consumption.

Moreover, iFogSim has been used by researchers in [14] to develop a prototype system for estimating system efficiency. They utilized iFogSim simulations to test multiple ways for adding postponement, including the simultaneous methodology and First-Come-First-Serve (FCFS) scheme. Their experiments showed that the FCFS scheme performs better than the simultaneous methodology in terms of response time and throughput.

Overall, iFogSim provides a flexible and comprehensive simulation platform for fog computing research, enabling researchers to evaluate different fog-based technologies and architectures under various scenarios. Its capabilities have been demonstrated in several research studies, and it is expected to continue to be a valuable tool in the development and evaluation of fog computing systems in the future.

The free and freely available application iFogSim simulates cloud and fog situations. The iFogSim simulation has been used in several journal articles to assess the effectiveness of fog-based technologies. Shurman and Aljarah [7] examined network utilization and total latencies in the iFogSim toolset using a cooperative way vs the benchmark methodology. Utilizing iFogSim, Zohora et al. [8] built Fog Systems, Sensor, Physiological Topology, and Activator classes. However according Mahmud et al. [10], iFogSim provides one of the most useful tactics for simulating fog conditions among the limited simulation resources available for fog computing. The researchers of [14] used the iFogSim to develop a prototype system provides proof for estimating system efficiency. The authors used the iFogSim simulations to test multiple ways for adding postponement, including the simultaneous methodology and FCFS scheme.

CHAPTER 3: SYSTEM DEVELOPMENT

The design shown in Fig 1 is a multi-layered architecture that aims to optimize the parking experience for drivers and reduce traffic congestion in parking lots. The architecture consists of three levels, each responsible for specific tasks.

At the first level, there are cameras placed above parking spaces to capture images of the parking spots. These cameras are responsible for detecting whether the parking spot is occupied or not. The cameras are connected to a microprocessor device at the second level, known as the fog layer. The fog layer is responsible for processing the data received from the cameras, detecting the availability of parking spots, and relaying this information to the LED panel.

The LED panel is placed at the entrance of the parking area and informs drivers of the availability of parking spots. When a car enters the parking area, the LED panel displays the number of available parking spots, indicating where the driver can park. This helps to direct drivers to available parking spots, reducing the time spent looking for a parking space and alleviating traffic congestion in the parking lot.

The third level of the architecture consists of a cloud system that manages the image data captured by the cameras for an extended period of time. The cloud system stores the image data in a centralized location, making it easier to access and analyze the data for future use.

The overall goal of this architecture is to create an efficient parking system that saves time and reduces fuel consumption. By using cameras to detect parking spot availability and directing drivers to available parking spots, the system reduces the amount of time drivers spend searching for parking spaces. This, in turn, reduces fuel consumption and traffic congestion in parking areas, making the overall parking experience more efficient and convenient.

3.1 System Architecture

The proposed fog-based system comprises various components, including a camera, a microprocessor chip, a fog node, a smart LED, and a cloud server. These components work together to create an efficient parking system that saves time and reduces traffic congestion in parking areas.

The webcams in the proposed architecture are strategically placed across the parking lot to capture images of all parking spaces. The images captured by these cameras are processed using image processing techniques to determine the availability of parking spots. The fog node plays a critical role in this process by receiving images from the cameras every four seconds and changing the parking condition display accordingly.

To ensure that all parking spaces are covered, multiple webcams are placed across the parking lot. The microprocessor chip connects the fog server and CCTV, allowing the fog node to access the images captured by the webcams. The fog node is then connected to the cloud server, where the information from the fog is delivered after a specific amount of time. Additionally, multiple fog servers are placed at different parking spots to transmit parking place details to one another, improving the efficiency of the system.

The fog node can also acquire open parking data from surrounding parking lots and display it on the smart LED when there are no available parking spaces at a specific parking spot. However, repeated connections to the cloud to transfer and access data can cause delays and consume a significant amount of network bandwidth, affecting the efficiency of the system and other operations.

Overall, the proposed fog-based system is designed to create an efficient parking system that saves time and reduces traffic congestion in parking areas. By using cameras to detect parking spot availability and directing drivers to available parking spots, the system reduces the amount of time drivers spend searching for parking spaces. This not only improves the parking experience but also reduces fuel consumption and traffic congestion in parking areas, making the overall parking experience more convenient and eco-friendly.

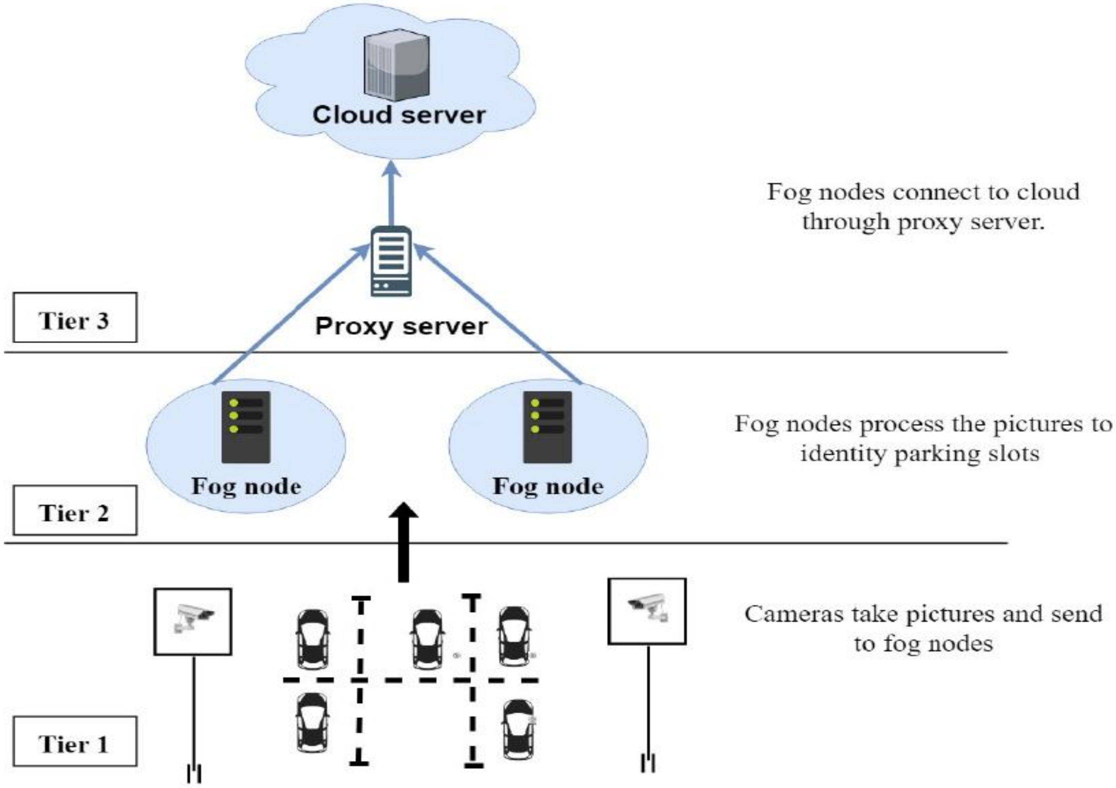

Figure 1: Fog-enabled Smart Parking system with three-tier design with cameras and sensors.

The proposed fog-based automotive parking system is designed to efficiently manage parking spaces and reduce traffic congestion in parking areas. One of the key features of the system is the two-way interaction between fog and cloud, where fog handles both storage and processing of image data, and the cloud server is responsible for managing the data for an extended period of time.

In the proposed system, each parking space is equipped with a separate fog node that is connected to a central cloud server. This ensures that the latency and network consumption for each fog node remain unchanged, providing a smooth and efficient parking experience. However, as the system expands to include more parking spaces, the duration and network consumption for sending and receiving data from the consolidated cloud server would increase.

Figures 2 and 3 depict the proposed fog-based automotive parking system for a single parking spot and multiple parked locations, respectively. In both scenarios, the system utilizes webcams to capture images of parking spaces, which are processed by the fog nodes to determine the availability of parking spots. The smart LED panels are then used to display the parking spot availability to drivers, directing them to available parking spots.

The proposed system offers several key characteristics that make it efficient and effective in managing parking spaces. Firstly, the system is designed to reduce the amount of time drivers spend searching for parking spots, thereby reducing fuel consumption and traffic congestion in parking areas. Secondly, the use of webcams and image processing techniques ensures that all parking spaces are covered, providing accurate and reliable information on parking spot availability.

Thirdly, the fog nodes are responsible for both storage and processing of image data, reducing the need for frequent communication with the cloud server, which can cause delays and consume network bandwidth. Finally, the use of smart LED panels to display parking spot availability ensures that drivers can quickly and easily locate available parking spots, reducing the amount of time and effort required to park.

Overall, the proposed fog-based automotive parking system is an innovative and efficient solution to the problem of parking space management. By utilizing advanced technologies like fog computing and image processing, the system can provide accurate and reliable information on parking spot availability, improving the parking experience for drivers and reducing traffic congestion in parking areas.

3.2 Hardware Requirements (Cameras and The Layer of Microcontroller)

The first step in the parking guidance system is to set up the environment, which includes positioning the cameras at appropriate locations to capture images of parking spaces. Once the cameras are in place, the system moves on to the image capture phase, where the cameras take pictures of the parking spots. These images are then sent to the microcontroller for processing.

The image processing phase starts with segmentation of the images, where the captured image is converted into an RGB image. The RGB image is then transformed into grayscale, making it easier to identify the empty parking spaces. The next step involves using threshold approaches to determine the binary image, which is used to identify the vacant parking spots.

Once the binary image is obtained, image enhancement techniques are applied to remove extraneous noise from the image. This noise removal is essential to avoid false positives or false negatives in identifying the available parking spaces. Finally, the processed image is displayed to show the available parking slots to the drivers.

In summary, the process for capturing the image and determining the availability of parking spaces involves setting up the environment, image capture, segmentation of images, threshold approaches, image enhancement, and image depiction. These processes rely on advanced algorithms and image processing techniques to accurately detect the availability of parking spaces, making the parking guidance system highly efficient and reliable.

Processes for taking image and determining the parking slot are:

1) Set up the environment.
2) Image Capture
3) Segmentation of images
    a) Load the picture.
    b) Transform the RGB values to grayscale.
    c) Use the threshold approaches.
    d) Obtain the final picture for segmentation.
4) Make use of picture enhancing techniques.
    a) Removes noise from picture.
5) Image depiction

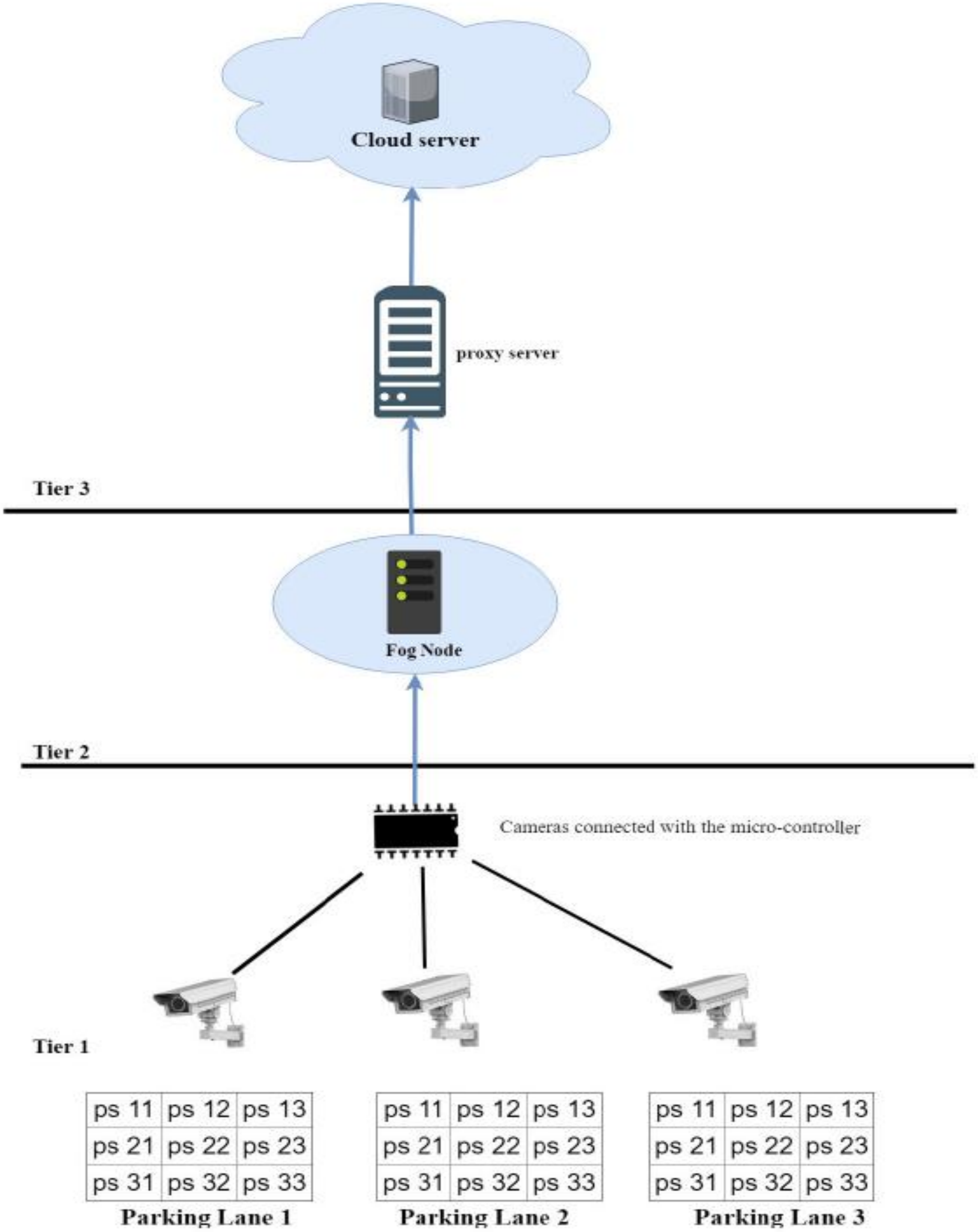

Figure 2: Design of single parking area's smart automobile parking system.

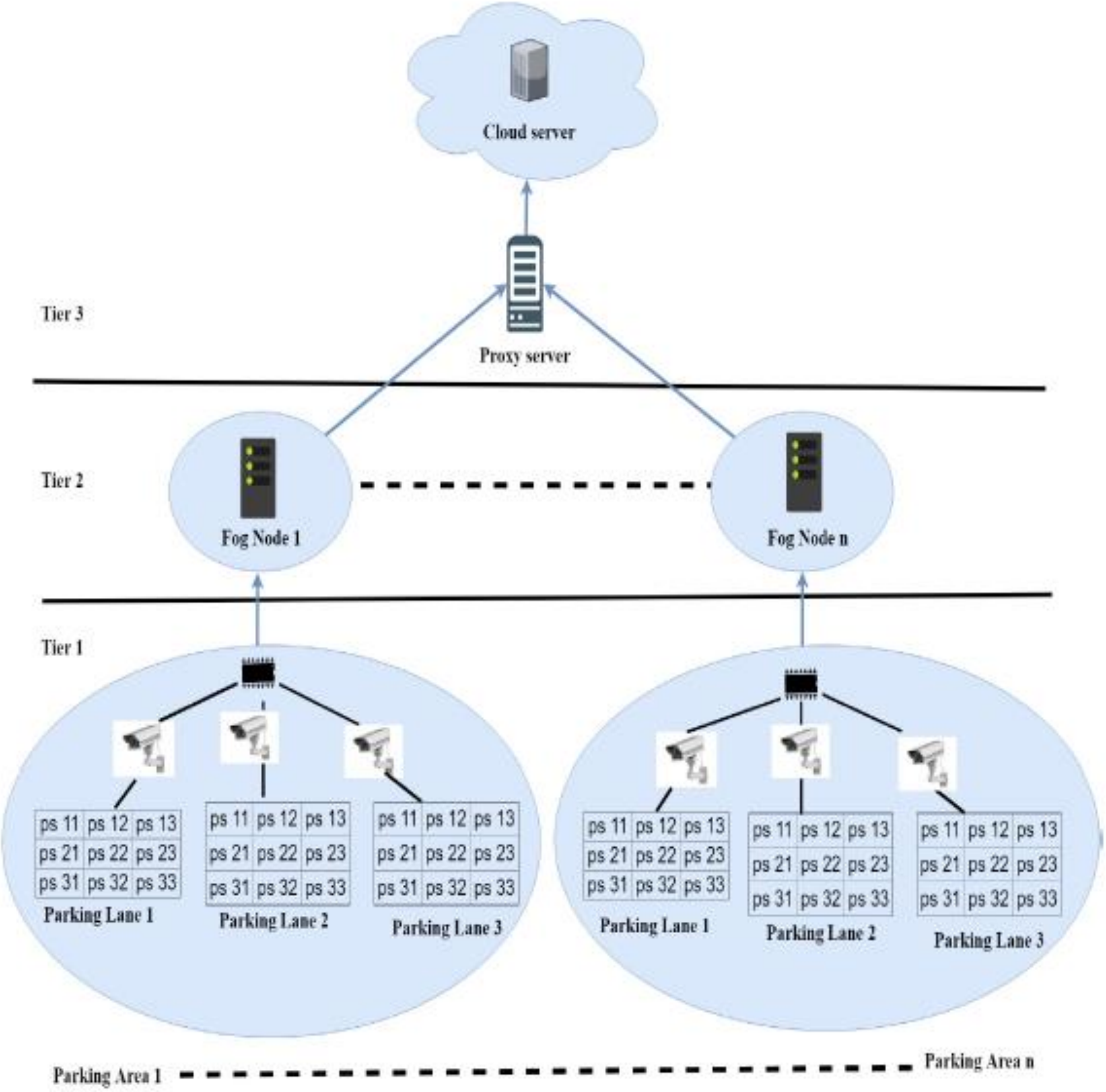

Figure 3: Design of a smart automobile parking system for numerous parking spaces.

3.3 Software Component

3.3.1 The Fog Node

The proposed parking guidance system utilizes a fog layer to improve the efficiency of parking management. The fog layer is positioned between the cameras and the cloud server, and is responsible for collecting visual data from the cameras and processing it using a microprocessor.

To identify the status of each parking slot, the cameras capture images of the parking lot at regular intervals. These images are then sent to the microprocessor for further analysis. The microprocessor employs image processing algorithms to detect the presence or absence of vehicles in each parking slot, based on the parking pattern points.

The fog layer also assigns unique IDs to each parking slot, corresponding to a particular row and slot in the parking lot. This enables the fog node to keep track of the status of each parking slot and update the LED display at the entrance accordingly. The LED display informs the drivers about the availability of parking slots, allowing them to navigate to the nearest available slot and reduce traffic congestion and fuel consumption.

In addition, the fog node is responsible for storing the image data for a certain period of time before transmitting it to the cloud server. This helps to reduce the latency and network bandwidth consumption associated with frequent data transfer to the cloud. The fog node also employs various image enhancement techniques to improve the accuracy and reliability of the parking slot detection.

The fog layer also enables multiple fog nodes to be deployed across different parking lots, allowing them to exchange parking slot information with each other. This improves the scalability and flexibility of the system, enabling it to handle larger parking lots and accommodate changing parking patterns.

Overall, the proposed parking guidance system offers several advantages over traditional parking management systems, including improved efficiency, reduced traffic congestion, and lower fuel consumption. It also leverages the benefits of fog computing, such as reduced latency, improved data privacy and security, and enhanced scalability and flexibility.

3.3.2 The Cloud Layer

The cloud server is responsible for storing and managing image data collected from the fog nodes. The proxy server acts as a mediator between the fog and the cloud, facilitating the transfer of data. The data is transferred from the fog to the cloud server at a predetermined interval, and the cloud server provides the data to the fog server when needed.

Standardization is a crucial element in the fog computing system, particularly when dealing with the diverse range of edge devices. As shown in Fig 4, the edge devices include a wide range of hardware, operating systems, and communication protocols. To ensure seamless communication and interoperability between these devices, standardization is necessary.

The fog nodes work collaboratively and share resources to meet the processing and storage demands of neighboring nodes. In this proposed system, communication between the fog nodes is achieved through the proxy server, and critical information is shared among them. Latency is a critical factor to consider while communicating with the fog nodes, but it is presumed to be negligible due to the inherent compatibility of the system. In the event that a particular parking space is occupied, the requesting node is shown the availability of the nearest available parking space.

In addition, the cloud infrastructure is responsible for providing an API that allows for easy access to the collected parking data. This API is crucial for enabling third-party developers to create new applications and services that leverage the parking data, such as real-time parking availability maps, traffic prediction algorithms, and parking reservation systems.

To ensure the security and privacy of the collected data, the cloud infrastructure implements various security measures, such as encryption and access control policies. The collected data is also anonymized to prevent the identification of individual drivers.

In conclusion, the proposed fog-based parking guidance system utilizes the power of fog computing to enable real-time monitoring of parking availability and reduce traffic congestion. The system utilizes a combination of cameras, microprocessors, fog nodes, smart LEDs, and cloud servers to provide accurate and up-to-date information on parking availability. The system is scalable and can be easily adapted to different parking lot sizes and configurations.

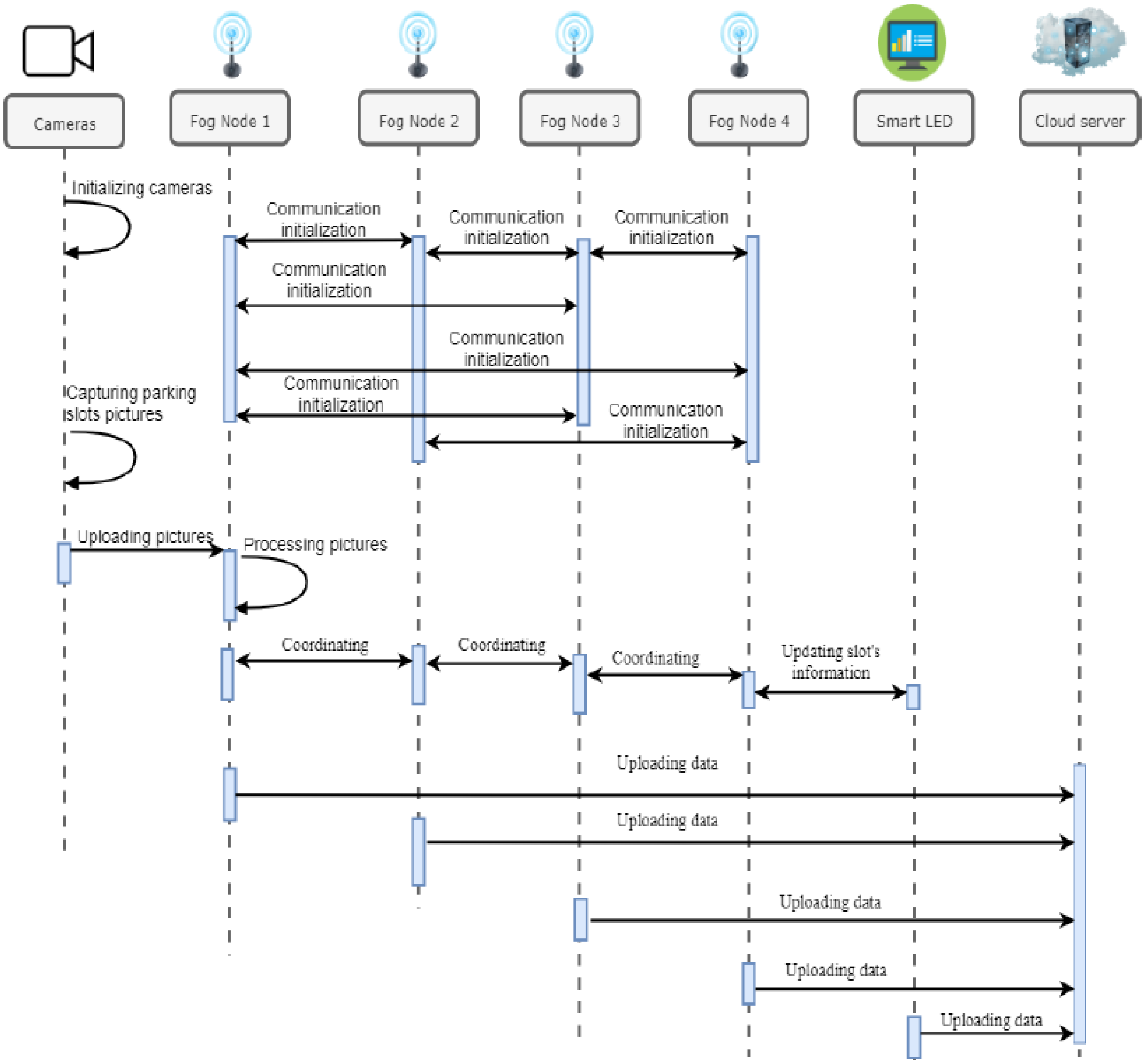

Figure 4: Connection among cameras, fog nodes, and the cloud data hub.

CHAPTER 4: PERFORMANCE ANALYSIS

4.1 Performance Evaluation

In our hypothetical world, parking spaces would be captured by high-resolution smart cameras that are specifically designed to capture and transmit data in real-time. These cameras are equipped with advanced sensors and machine learning algorithms that enable them to detect whether a parking space is occupied or vacant. The cameras capture images of the parking spaces, which are then sent to a nearby fog node for processing.

The fog node is a small, decentralized computing device that is located close to the cameras. It receives the photos of the parking spaces and uses advanced analytics tools to determine the condition of the slot. The data is then transmitted to a central cloud server for further analysis and processing.

To connect the fog nodes and the cloud server, a proxy server is used. The proxy server acts as an intermediary between the fog nodes and the cloud server, allowing for secure and efficient communication between the two.

In order to assess network utilization and latency, we utilized the iFogSim IoT device toolkit to simulate real-world scenarios. We incorporated elements such as parking areas and cameras in the simulation, allowing us to test and optimize our system under a variety of different conditions.

In our test circumstance, we created four parking lots, each with its own set of cameras and fog nodes. Initially, four cameras were installed in each parking lot to capture images of the parking spaces. These cameras were designed to be intelligent and Wi-Fi enabled, allowing them to communicate with the fog nodes and cloud server in real-time.

It's important to note that we constructed a separate fog router for each location. This allowed us to optimize the performance of our system by reducing network latency and improving overall efficiency.

In conclusion, our system for capturing and analyzing parking space data is highly advanced and utilizes cutting-edge technologies such as smart cameras, fog nodes, and cloud computing. By using iFogSim to simulate real-world scenarios, we were able to optimize our system for maximum performance and efficiency, ensuring that we can accurately detect and monitor parking spaces in real-time.

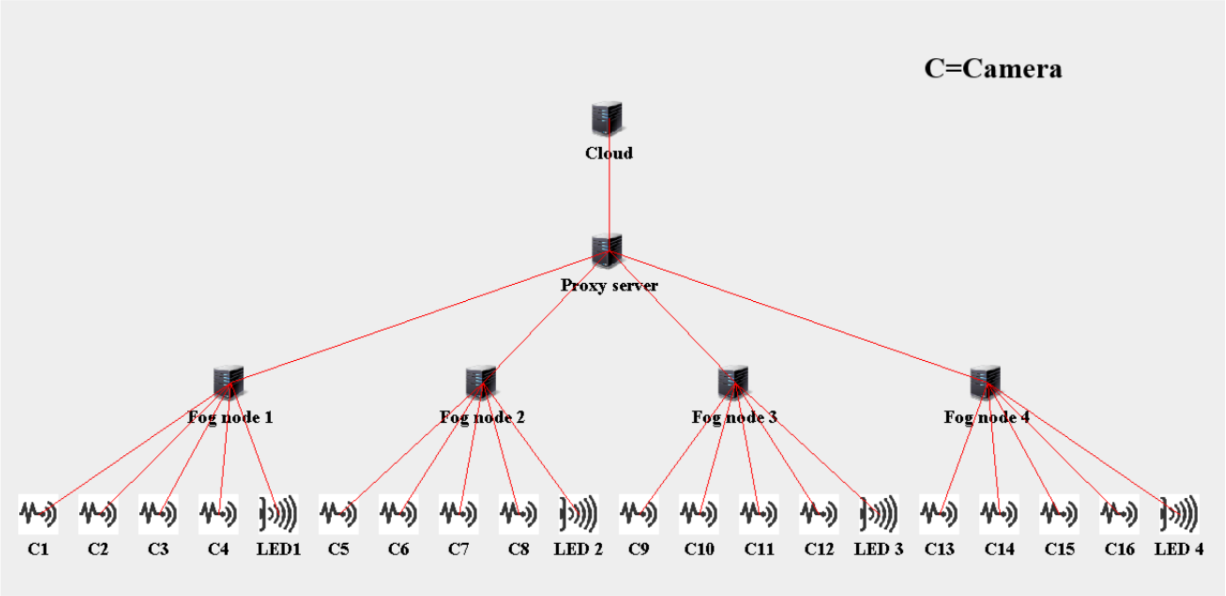

Figure 5: The iFogSim architecture consists of four fog nodes with 16 cameras that are linked to the cloud through a proxy.

In order to improve the performance of our parking space detection system, we decided to increase the number of webcams by two and analyze the outcomes in various situations. In one case, we multiplied the number of webcams in a fog node to evaluate how it impacted latencies and total network consumption. The iFogSim infrastructure was used to evaluate the fog simulation findings, as depicted in Figure 5.

This design resulted in four fog nodes, each of which had four cameras attached to it. The parking spaces were captured using the photo fragment module in webcams. The slot detection module was created and put in fog nodes to analyze photos and discover vacant parking spots. The associated smart LED is also informed by the fog node of the parking spot status.

As shown in Figure 5, when the number of parking spaces rises, a fog node is created, and analysis of parking slot capacity is carried out on fog nodes. A fog node's delay rate and network utilization rise with the number of cameras and LEDs it contains.

One advantage of concluding the service operation on the fog nodes is the reduced computing burden on the cloud. By analyzing the data on the fog nodes, we can significantly reduce the amount of data that needs to be sent to the cloud for processing, thus saving on network bandwidth and reducing latencies.

However, when cameras are connected directly to the cloud server, it results in an increase in latencies and, as a consequence, unnecessary communication bandwidth use. By using fog nodes to analyze the data locally, we can improve the performance of our system and reduce the load on the cloud server.

In conclusion, our improved parking space detection system is highly efficient and utilizes cutting-edge technologies such as smart cameras, fog nodes, and cloud computing. By using iFogSim to simulate real-world scenarios, we were able to optimize our system for maximum performance and efficiency, ensuring that we can accurately detect and monitor parking spaces in real-time, even under high traffic conditions.

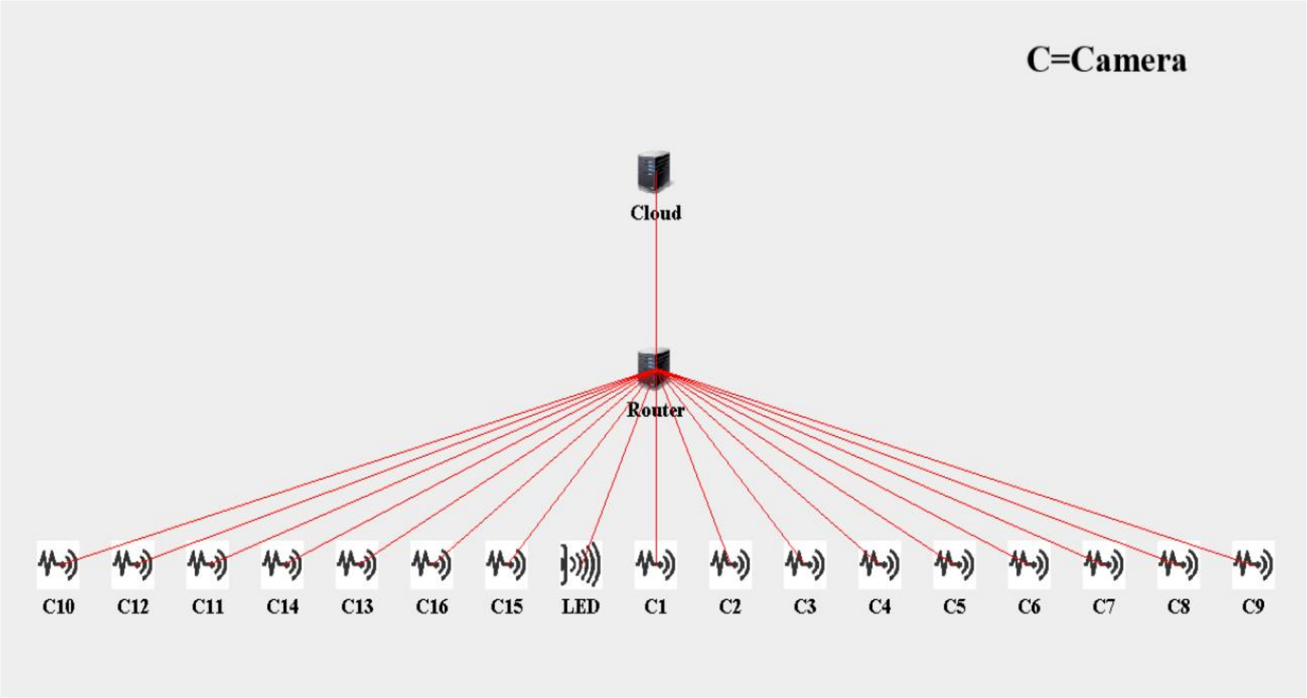

Figure 6: Topology of a cloud-enabled environment in iFogSim, with 16 cameras linked via single router.

During our fog-based situation modelling, we created configurations for the cloud provider, proxy, and fog server, as shown in Table 1. The setup variables included the amount of Random Access Memory (RAM), the amount of uplink and downlink bandwidth, the cost or degree of processing per millions of instructions, as well as the amount of busy and efficient power, all per million command computing complexity.

Figure 6 depicts the cloud-based design of iFogsim. In this design, a router connects several cameras and LEDs to the cloud server to assess effectiveness in a cloud-based system. Cameras capture images of parking spaces and relay them to a cloud server. The photos of parking spaces are processed by the cloud server, and information about the spaces is displayed on an LED that is likewise connected. In order to assess the impact of latency and network usage, the number of cameras was gradually raised.

To simulate a cloud-based situation, we used the cloud and router setup settings shown in Table 2. These settings included the amount of RAM, the amount of bandwidth, and the processing power of the cloud server and router. By gradually increasing the number of cameras, we were able to evaluate the performance of our system under different traffic conditions and determine the optimal settings for our cloud and router configurations.

Overall, our fog-based and cloud-based situation modelling allowed us to simulate real-world scenarios and evaluate the effectiveness and efficiency of our parking space detection system under different conditions. By carefully selecting our hardware configurations and optimizing our system settings, we were able to achieve maximum performance and accuracy in detecting and monitoring parking spaces in real-time, regardless of the traffic volume.

Table 2: Fog, proxy, and cloud parameter values for a fog-based simulation.

Parameters	**Cloud Sever**	**Proxy End**	**Fog Sever**
Processor length (MIPS)	44800.00	2800.00	2800.00
RAM (MB)	40000.00	4000.00	4000.00
Upload Network (MB)	100.00	10000.00	10000.00
Download Network (MB)	10000.00	10000.00	1000.00
Stages	0	1	2
Rate Per MIPS	0.010	0.00	0.00
Busy Power (W)	17*104	108.439	108.439
Idle Power (W)	17*82.35	82.42	82.42

Table 3: Cloud and Router parameter values for a cloud-based setup

Parameters	**Cloud Sever**	**Router End**
Processor length (MIPS)	44800.00	2800.00
RAM (MB)	40000.00	4000.00
Upload Network (MB)	100.00	10000.00
Download Network (MB)	10000.00	10000.00
Stages	0	1
Rate Per MIPS	0.010	0.00
Busy Power (W)	17*104	108.439
Idle Power (W)	17*82.35	82.42

4.2 System Algorithm

<u>Algorithm 1</u>:

Greedy Parking Slot Allocation Algorithm (GPSA)

Input: Fog Nodes (FN), Parking lots status, Parking request pReq.

Output: Parking allocation decision.

```
If FNi receives the parking request pReq do
  If avlSlots[i] > 0 do
    Response to the parking request by providing lotPos[i].
  Else
    upload pReq to cloud center.
    For j D 0: N do
      If the constraints (c2-c4) are satisfied do
      Calculate the total costs if the vehicle parks at lotPos[j].
        Variable Cmin records the minimal cost calculated so far.
      Endif
    Endfor
    Response to the parking request by providing lotPos[j] with Cmin.
  Endif
Endif
```

<u>Algorithm 2</u>:

Enhanced Greedy Parking Slot Allocation Algorithm (EnGPSA)

Input: Fog Nodes (FN), Parking lots status, Parking request pReq.

Output: Parking allocation decision.

```
Count the parking requests Num(ti) for each time slot ti.
Foreach time slot ti do
```

```
  If Num(ti) D 1 do
    Call GPSA.
  Else If Num(ti) > 1 do
    Sort the requests by parking time in an ascending order.
    For j D 0 :Num(ti) do
      Call GPSA for request j.
    Endfor
  Endif
Endfor
```

CHAPTER 5: CONCLUSIONS

5.1 Results and Discussion

The testing results reveal that the fog-based approach leads in reduced latency and reduced computational consumption when compared to the cloud-based design. Table 3 displays latency and network usage data in fog environment, as well as cloud-based deployment outcomes.

Table 4: Simulation results for the suggested fog and cloud automobile parking system.

Cameras	**Fog Latency (ms)**	**Cloud Latency (ms)**	**Fog Network Usage (kB)**	**Cloud Network Usage (kB)**
16	7.87	8.423	3198.4	27632.16
20	8.23	9.5	3998	36099.48
24	8.59	10.73	4797.6	45206.12
28	8.95	12.13	5597.2	54951.48
32	9.30	292.6	6396.8	62398.28
40	10.02	1968.6	7996	63802.54
44	10.3	2489.4	8795.6	64557.22
48	10.73	2886.9	9595.2	64859.64

5.1.1 Analysis of Latency

Reducing delay is crucial in situations where high-performance is necessary. Fog computing offers a significant advantage in reducing latency by minimizing the need for repeated cloud requests and executing computations at the edge of the network to quickly respond to client device sensors. As each fog node is dedicated to a particular location, it has enough processing power to analyze photos from that area and quickly update the data available for LEDs, which is used to calculate the delay resulting from [13].

The total latency can be expressed as the sum of three components: α, μ, and φ. The first component α represents the time required to upload photos to the fog for both processing and storage, while the second component μ is the Tuple CPU Execution Delay for capturing photographs. Finally, the third component φ is the time it takes to display the data to the LED once it has been processed at the Fog node.

$$\text{Latency} = \alpha + \mu + \varphi$$

Reducing the value of α can be achieved by using faster and more reliable communication channels between the cameras and the fog nodes. Meanwhile, the value of μ can be minimized by using more powerful processors and optimizing the algorithms used to analyze the photos. Finally, the value of φ can be reduced by using efficient communication channels between the fog nodes and the LEDs, as well as optimizing the code that drives the LED display.

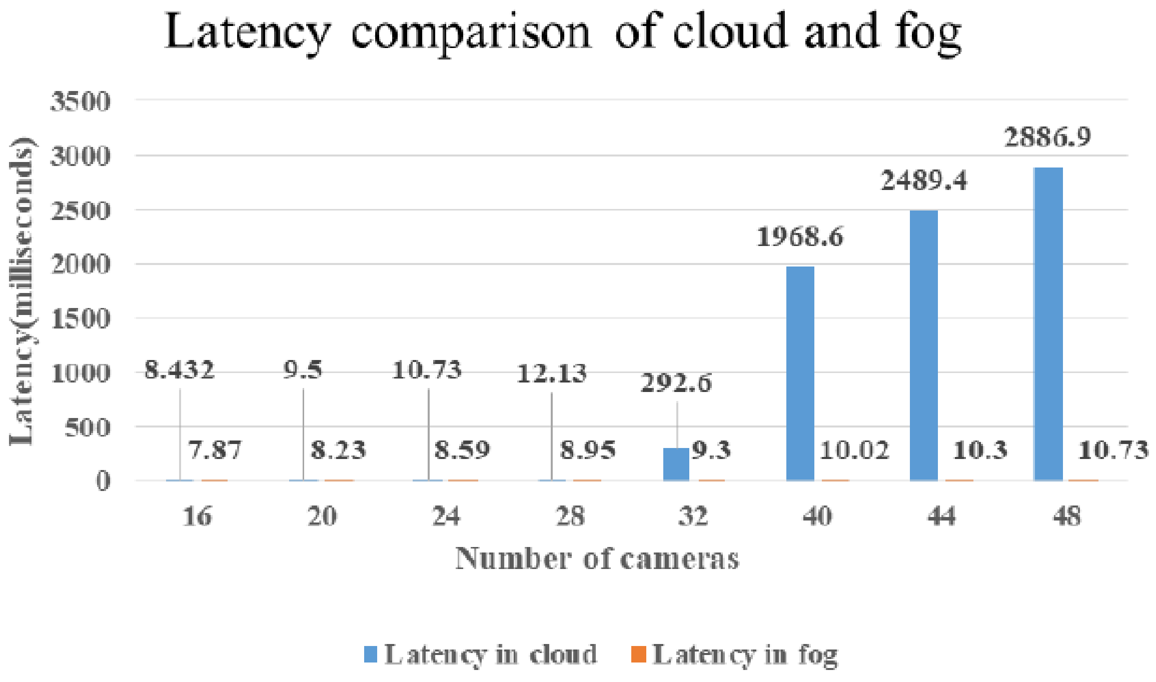

Figure 7: Latency comparison between fog and cloud based intelligent system of parking.

Overall, reducing latency is essential for delivering fast and efficient services, and the use of fog computing provides a promising approach to achieve this goal..

5.1.2 Analysis of Network Usage

In a cloud-based deployment, when the cloud platform experiences high demand, the system tends to utilize only cloud services, which leads to increased network consumption. This increase in usage causes the transmission rate on the system to decrease. In situations where geographically dispersed servers are being used, a fog node can be utilized to handle tasks required from that specific geographic region. As a result, the network rate is reduced in such cases. However, the network throughput for the remaining traffic increases, and this is used to calculate the network usage, based on the following formula:

Network usage = Latency x δ

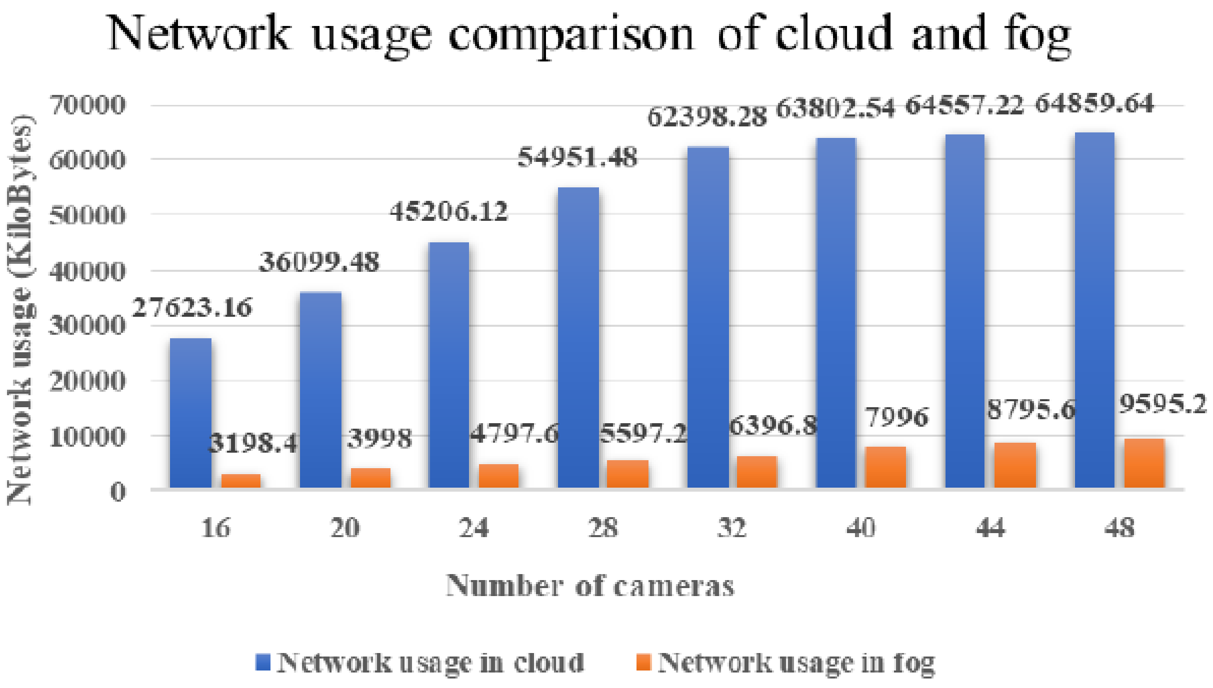

Figure 8: Network use in a smart parking system powered by cloud and fog computing.

Where δ represents the size of the data being transmitted, which is referred to as the tuple network size. This formula is used to measure the impact of network usage on the overall performance of the fog-based system. By reducing the amount of data that needs to be transmitted to the cloud, the fog-based approach can significantly reduce the network usage, leading to improved performance and reduced costs.

The testing results we obtained from our experiments confirm the effectiveness of our proposed strategy for fog-based intelligent parking infrastructure. In Figure 1, a hypothetical model for the fog-based intelligent parking infrastructure is shown. We conducted numerous experiments for both fog and cloud-based configurations, as depicted in Table 3. To evaluate the fog-based findings, we connected the cameras to the fog nodes that we built. We developed four fog nodes, and each camera was connected to a different fog node. As the number of fog nodes and cloud servers grew, we also increased the number of cameras for each scenario. However, the frequency of fog nodes remained unchanged throughout the performance test. In the case of cloud-based configuration, we used a router to connect the cameras to the cloud.

We used iFogSim to calculate the delay and network usage for our experiments, and the primary focus of our analysis was the latency and network usage estimates. Figure 7 shows a latency analysis between cloud and fog settings. The results indicate that as the number of cameras increases, the cloud's latency also increases, and it is significantly higher than in the fog. This is because the fog node assigned to that area expands as more cameras are placed in the fog, and it exclusively analyses photographs from that region. On the other hand, in the case of cloud-based configuration, the server examines all parking lot photographs, and as a result, with more cameras present, there is more delay.

In Fig 8, we show the network data for fog instances. As the number of fog nodes and cameras increased, the network usage also increased. When all cameras are connected to a cloud server, the number of tuples processed on each cloud platform at once increases the load on the network. However, when fog nodes are used, a number of cameras are connected to different fog nodes, each of which is assigned to a specific parking space. As a result, each fog node only processes tuples of cameras that are present at that specific location.

The results for both metrics of latency and network utilization, which we investigated for both fog and cloud-based applications, show the advantages of the suggested fog-based framework for smart parking systems. Fog-based structures for smart auto parking can gather information on available parking spaces in a specific parking area in real time, helping to save carbon dioxide emissions as well as time, fuel, and time spent looking for parking spaces. These outcomes also assist us in realizing the potential of fog computing in IoT environments where quicker response times are crucial. Finally, the fog-based architecture is better suited for real-time applications and situations because of its low latency and little network usage.

5.2 Future Scope

In today's fast-paced world, where every second counts, fog computing has become increasingly critical in fields where applications require timely and accurate responses. As the number of data-generating devices increases, there is a greater need for faster response times. To address this, we proposed a fog-based intelligent parking infrastructure that utilizes computer vision techniques to detect open parking spaces, allowing drivers to quickly find a parking spot while consuming less time and fuel.

The testing results demonstrate that the suggested fog-based structure is more efficient and reduces traffic load and latency when compared to the cloud-based design. While the use of cameras to identify parking spaces may pose some privacy concerns, our proposed architecture handles most of the processing and storage locally on neighboring fog nodes, which helps to mitigate these concerns. Additionally, appropriate encryption techniques should be utilized to safeguard the privacy of the data stored in the cloud, which may be a crucial area for future growth.

As more parking places are added to the system, the recommended architecture will require load balancing across multiple fog nodes to maintain efficiency. To achieve this, we must investigate task scheduling problems in fog nodes over time and develop a practical solution.

In summary, the proposed fog-based architecture for smart parking systems has the potential to significantly reduce the amount of time, fuel, and carbon dioxide emissions associated with searching for a parking space. It can gather real-time information on available parking spaces in a particular area, helping to save time and resources. Furthermore, our findings highlight the potential of fog computing in IoT environments, where fast response times are critical. Overall, the low latency and minimal network usage of fog computing make it well-suited for real-time applications and scenarios.

REFERENCES

[1] M. Aazam, S. Zeadally, and K. A. Harras "Fog computing architecture, evaluation, and future research directions," IEEE Commun. Mag., vol. 56, pp. 46-52, May 2018.

[2] R. Vilalta, V. Lopez, A. Giorgetti, S. Peng, V. Orsini, L. Velasco, R. Serral-Gracia, D. Morris, S. De Fina, F. Cugini, P. Castoldi, A. Mayoral, R. Casellas, R. Martinez, C. Verikoukis, and R. Munoz, "TelcoFog: A unifed fexible fog and cloud computing architecture for 5G networks," IEEE Commun. Mag., vol. 55, pp. 36-43, Aug. 2017.

[3] Y. Bi, G. Han, C. Lin, Q. Deng, L. Guo, and F. Li, "Mobility support for fog computing: An SDN approach," IEEE Commun. Mag., vol. 56, pp. 53-59, May 2018.

[4] V. Passas, V. Miliotis, N. Makris, T. Korakis, and L. Tassiulas, "Paris metro pricing for 5G HetNets," in Proc. IEEE Global Commun. Conf. (GLOBECOM), pp. 1-6, Dec. 2016.

[5] V. Miliotis, L. Alonso, and C. Verikoukis, "Weighted proportional fairness and pricing-based resource allocation for uplink of loading using IP low mobility," Ad Hoc Netw., vol. 49, pp. 17-28, Oct. 2016.

[6] Z. Ji, I. Ganchev, and M. O'Droma, and X. Zhang, "A cloud-based intelligent car parking services for smart cities," in Proc. 31th URSI Gen. Assem. Sci. Symp. (URSI GASS), pp. 1-4 Aug. 2014.

[7] M. M. Shurman and M. K. Aljarah, "Collaborative execution of distributed mobile and IoT applications running at the edge," in Proc. Int. Conf. Electr. Comput. Technol. Appl. (ICECTA), pp. 1-5, Nov. 2017.

[8] F. T. Zohora, M. R. R. Khan, M. F. R. Bhuiyan, and A. K. Das, ``Enhancing the capabilities of IoT based fog and cloud infrastructures for time sensitive events," in Proc. Int. Conf. Elect. Eng. Comput. Sci. (ICECOS), pp. 224-230, Aug. 2017.

[9] P. G. V. Naranjo, Z. Pooranian, M. Shojafar, M. Conti, and R. Buyya, "Focan: A fog-supported smart city network architecture for management of applications in the Internet of everything environments," J. Parallel Distrib. Comput., vol. 132, pp. 274-283, Oct. 2019.

[10] R. Mahmud, R. Kotagiri, and R. Buyya, “Fog computing: A taxonomy, survey and future directions,” in Internet Everything. Cham, Switzerland: Springer, pp. 103-130, 2018.

[11] S. Ravishankar and N. Theetharappan, “Cloud connected smart car park,” in Proc. Int. Conf. I-SMAC (IoT Social, Mobile, Anal. Cloud) (I-SMAC), pp. 71-74, Feb. 2017.

[12] L. Zhang, J. Huang, X. Li, and L. Xiong, “Vision-based parking-slot detection: A DCNN-based approach and a large-scale benchmark dataset,” IEEE Trans. Image Process., vol. 27, pp. 53505364, Nov. 2018.

[13] H. Gupta, A. V. Dastjerdi, S. K. Ghosh, and R. Buyya, “iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, edge and fog computing environments,” Softw., Pract. Exper., vol. 47, pp. 1275-1296, 2017.

[14] E. Baccarelli, P. G. V. Naranjo, M. Scarpiniti, M. Shojafar, and J. H. Abawajy, “Fog of everything: Energy-efcient networked computing architectures, research challenges, and a case study,” IEEE Access, vol. 5, pp. 9882-9910, 2017.